The main logic is in the void loop() and void countdown(). After the void setup() is completed the control comes to the void loop(). Initially when you power the Arduino the count time value is zero(there is nothing to count down). So, we have to set some time and for that we will press the position button. Now using the position button you can also set the minutes and seconds(If ‘Num’ value is even we set the minutes and if it is odd we set the seconds). Using the increase & decrease button we can adjust our time. After the time is set by us we will calculate the total time in ‘seconds’ using this formula (count\_time = left\_value\*60 + right\_value).

Let’s say we have set the count down time of 30 seconds. Now, we will start the timer and for that we will press the ‘Start’ button. After pressing the start button,

if(start\_pause == 1) condition will get satisfied and the

function named ‘countDown’ will get called. The ‘countDown’ function does the main job of decrementing the timer by one second and updating the TM1637 display. The start/pause button in this project is used as ‘hardware interrupt’ i.e whenever you press this button the Arduino will stop doing its work and will go in the function called ISR(check\_button in this case) and return back to whatever it was doing after executing the code written in the ISR.

This is how whenever you press the start/pause button the value of

‘start\_pause’ gets updated.

NOW, WE ARE IN THE FUNCTION ‘COUNTDOWN’

if(current\_time - previous\_time >= 1000)

The above line of code will look that the display is getting updated after every one second.

seconds = (NewTime % 60);

minutes = ((NewTime / 60) % 60 );

The above two lines of code will separate how much minutes and seconds are there from the given time. Before calling the function ‘countDown’ we have stored the time in seconds in the variable count\_time by using count\_time = left\_value\*60 + right\_value.

After the timer has finished counting, count\_time > 0 condition will not be true anymore and we will be out of the while loop and back to the point were the function was called. Again in the void loop() we will wait for any button press and perform the actions accordingly.

There is only one button here to start/pause the timer. So, if you press the pause button while the count down is happening the current/present time value will get stored and it will be displayed constantly until and unless you press the start button again.